Theses and Dissertations   1. Thesis and Dissertation Collection, all items

1989-09

A weapons systems development decision support system

Boyd, Richard K.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/38543

AD-A251 791

||||||||||||||||||||||||||||

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A WEAPONS SYSTEMS
DEVELOPMENT
DECISION SUPPORT
SYSTEM

by

Richard K. Boyd

March 1992

Thesis Advisor: Gordon Nakagawa

92-15589

|||||||||||||||||||||||||||||

92 6 1

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Computer Science Dept. Naval Postgraduate School	6b. OFFICE SYMBOL (if applicable) AS	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School

6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
A WEAPONS SYSTEMS DEVELOPMENT DECISION SUPPORT SYSTEM (U)

12. PERSONAL AUTHOR(S)
BOYD, Richard K.

13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM ___ TO ___	14. DATE OF REPORT (Year, Month, Day) March 1992	15. PAGE COUNT 75

16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Decision Support Systems, Multi-Attribute Decision Making, Hypertext, Weapons Systems Architectures

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
Defense budget cuts and the recent "peace dividend" have made weapons systems development decisions increasingly more difficult and subject to scrutiny. Meticulous planning is required to ensure tax dollars are spent wisely and effectively. This thesis presents a decision support system designed to aid a senior official in making such investment decisions. The system combines a graphical user interface embedded in a hypertext environment with a multiple attribute decision making method. Architectures, consisting of weapons systems development projects from each major program within a warfare area, which provide the best overall benefit versus cost are presented as solutions. The hypertext interface allows convenient access to benefit and cost data, and easily displays the solutions generated by the multiple attribute decision method.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [x] UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Gordon Nakagawa	22b. TELEPHONE (Include Area Code) (408) 646-2303 22c. OFFICE SYMBOL OR/NA

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted SECURITY CLASSIFICATION OF THIS PAGE
All other editions are obsolete

A WEAPONS SYSTEMS
DEVELOPMENT
DECISION SUPPORT SYSTEM

by
Richard K. Boyd
Lieutenant Commander, United States Naval Reserve
B.S. , U. S. Naval Academy, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1992

Author:

Richard K. Boyd

Approved By:

~~Gordon Nakagawa,~~ Thesis Advisor

Hemant Bhargava, Second Reader

David R.Whipple, Chairman,
Department of Administrative Sciences

ABSTRACT

Defense budget cuts and the recent "peace dividend" have made weapons systems development decisions increasingly more difficult and subject to scrutiny. Meticulous planning is required to ensure tax dollars are spent wisely. and effectively. This thesis presents a decision support system designed to aid a senior official in making such investment decisions. The system combines a graphical user interface embedded in a hypertext environment with a multiple attribute decision making solution method. Architectures, consisting of weapons systems development projects from each major program within a warfare area, which provide the best overall benefit versus cost are presented as solutions. The hypertext interface allows convenient access to benefit and cost data, and easily displays solutions generated by multiple attribute decision method.

iii

TABLE OF CONTENTS

I. INTRODUCTION

Major weapon system's development has become more expensive and subject to more careful scrutiny. A decision maker, faced with prioritizing development projects for funding approval, must increase his information sources and processing accordingly. Budget cuts and the recent "peace dividend" make those decisions even more important and difficult. Coupled with the long development times required by new or upgrade projects, careful, meticulous planning is necessary to ensure the ever-tightening budget dollar is allocated wisely. Historically, the U.S. Navy has divided its mission of maritime defense into several warfare areas. These are normally separated along natural borders relating to the medium in which the war is conducted, i.e. Anti-Air Warfare, Anti-Submarine Warfare, Anti-Ship Warfare, etc. [Ref.1] This approach lends itself to the selection of weapons systems development and procurement projects. Focusing on one warfare area reduces the weapons systems under consideration and conforms more closely to Congressional budget appropriations.

A. METHODOLOGY

This research paper will present a method and prototype decision support system to aid the decision maker in making fiscally responsible and informed selections regarding weapons

1

system's development and procurement. Application of this decision support system to current and future weapons systems planning decisions should present a more coherent and defendable acquisition policy to Congressional leaders.

The decision support system resides in a hypertext environment to allow the wealth of information on each option available to be digested in manageable portions. The information available on each option is obtained from various research groups within the existing Navy infrastructure. The expert opinions of these groups are made available for the decision maker to consider in support of the numeric assessments.

The intent of the project is to provide a briefing tool for the decision maker in a convenient format and on suitably portable hardware. The programming environment chosen was Hypercard. Hypercard allows a system designer to easily obtain powerful results and is supplied as standard software with every Macintosh. The numerical subsystem, written in C, was linked to Hypercard as an external resource. "What if" capabilities are provided to test the sensitivity of variances in the assessments made by expert research groups. Thorough justification data are available on an as-needed basis to allow the decision maker insight into the assessments and the subsequent impact on information provided.

B. RESEARCH QUESTIONS

The primary research question guiding this study is:

What is the best mix of Anti-Air Warfare development projects that will both maximize the capability and survivability of U.S. Navy assets given current budget constraints?

Subsidiary research questions addressed in this paper are:

1. What information is required for senior warfare decision makers to reach a best fleet mix?
2. How should this information be presented to the decision maker?
3. What method should be used to synthesize the raw data to produce the required information?

In order to address these questions, several research disciplines must be considered. Foremost of these are: Decision Theory; Hypertext; Multiple Criteria Decision Making Methodologies. Chapter II presents an introduction and literature review of these disciplines. Appendices A, B, and C contain more in-depth discussions of the theory. Chapter III addresses question (1). Chapter IV presents the overall design requirements and decisions made in implementing the decision support system. Finally, Chapter V presents conclusions and recommendations for further research, followed by selected exhibits from the system in Appendix D and E, a reference list, and bibliography.

II. THEORY AND LITERATURE REVIEW

A. DECISION SUPPORT SYSTEMS

Decision making processes can be characterized as ranging from strongly structured to completely unstructured. Structured decision problems occur when the methods to accomplish them are readily available, inputs are easily identified and the desired result is well defined. Simon theorized decision making as occurring in three phases: intelligence; design; choice. [Ref. 2] Unstructured decision problems exist when one or all three of the phases are not identifiable or standardized. Intuition is often the basis for making decisions on unstructured problems. [Ref. 3] Semi-structured problems fall somewhere in between the two previously mentioned, usually consisting of a well-defined solution method but requiring intuition to identify the desired result.

Decision making often follows predetermined strategies, resulting from the decision maker's preferences and the environmental pressures in force. Common decision making strategies are presented in Chankong and Haimes [Ref. 4].

Decision support systems (DSS), as envisioned by Gorry and Scott-Morton, [Ref. 5] are designed to aid in making semi-structured decisions. Ideally decision support systems improve the access to information through computerized

4

methods. ∪SS also include models of the decision environment. Often, a solver is provided in a DSS which conforms the data to the model and provides a solution, or several alternative solutions. [Ref. 6]

Another major characteristic of decision support systems is "what if" capabilities, a form of sensitivity analysis. To improve the decision maker's effectiveness, the DSS should be able to provide new solutions given alternate data sets. The value of computerized solution methods becomes manifest considering the speed and accuracy with which such systems can deliver results.

Sprague and Carlson theorized a generic design for decision support systems. [Ref. 7] Bonczek, Holsapple, and Whinston proposed a different, but similar design. In addition, Bonczek, et al, theorized a set of seven facets, common to all decision makers. These seven facets consist of three basic aspects and four attributes, which are combinations of the primary three. Figure 2-1 illustrates their relationships. Using these facets, Bonczek, et al, devised a method of evaluating a decision support system's "intelligence" based on the number of facets it automates. This intelligence provides a measure of the degree in which the DSS supports the decision maker. No DSS can fully replace a human decision maker, because all of his abilities cannot be automated. [Ref. 8] Appendix A summarizes these theories.

5
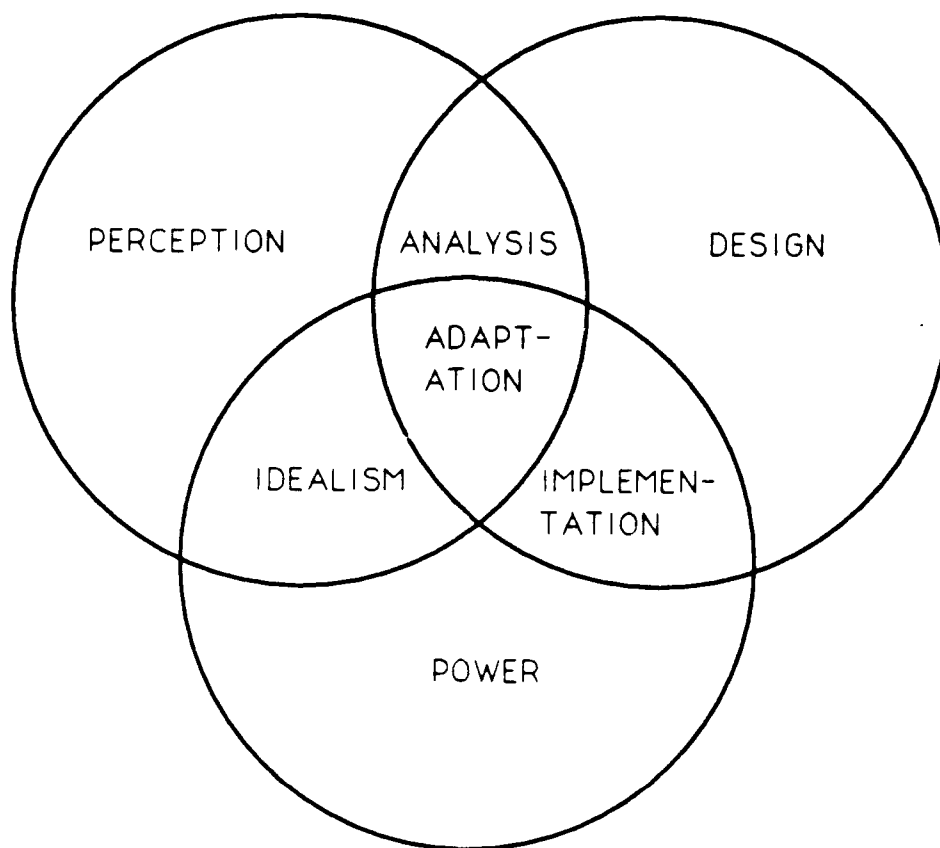
The Seven Facets of Decision Making

Three Aspects: Four Attributes:

POWER   ADAPTATION
PERCEPTION   ANALYSIS
DESIGN   IDEALISM
                                    IMPLEMENTATION

Figure 2-1

6

B. HYPERTEXT

Most naval warfare areas include several weapons systems, with each weapons system presenting a variety of options which may be chosen to upgrade or replace it. A plethora of data is available for the decision maker to assimilate into a meaningful form of information in order to make responsible decisions. An effective DSS will present this data for the decision maker in an easily managed interface to aid his decision process. One of the most popular and effective means of managing large volumes of data is the employment of hypertext.

Hypertext, coined by Ted Nelson, an early hypertext pioneer, is "a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ... of ... nonlinear text...." [Ref. 9] The literature is lacking in a more formal definition.

The hypertext concept consists of objects in a database which are linked together graphically and through pointers. The combination of these objects (nodes in the database) and their interconnecting links form a network called a hyperdocument. The linking feature provides the user (our decision maker) the "discretionary expansion of a document." [Ref. 10]

Employing hypertext allows the decision maker to gather the available data into manageable chunks, following the links provided by the designer. He may even create his own

links if they become meaningful for the decision at hand. This ability to dynamically link data into meaningful streams of information is believed by some researchers to closely approximate the operation of human associative memory. [Ref. 11] Conklin [Ref. 12] and Nielsen [Ref. 13] provide detailed descriptions of hypertext theory and usage. Summaries are discussed in Appendix B.

C. MULTIPLE CRITERIA DECISION MAKING

Real world decision making rarely incorporates only one criterion or goal. Attempts to force these decisions into single criterion models, oftentimes result in severe trivialization. The decision environment becomes so artificial that the model has little application. Multiple criteria models were created to represent actual decisions more realistically.

Multiple criteria decisions can be separated into two large categories based on the number of alternatives which must be considered. If a finite number of alternatives exist, the decision is often one of selection or evaluation. These decisions are considered as multiple attribute decisions. If the alternatives are infinite, the decision becomes one of design. These decisions lie in the area of research called multiple objective decision making. Hwang and Yoon [Ref. 14] provide a clear delineation of these distinctions. The objective of this study lies in the domain of multiple attribute decision making.

Multiple attribute decision making methods employ various models to simulate reality and associated solvers to provide the desired outcome. These models specify how the information on each attribute is processed. Two major models exist in multiple attribute decision making theory: noncompensatory and compensatory. [Ref. 14] Noncompensatory models do not permit tradeoffs between attributes. A decrease in the benefit provided by one attribute can not be offset by a corresponding increase. Compensatory models do permit these tradeoffs. As a result, compensatory model solvers are, in general, more complex then their counterpart solvers for noncompensatory models.

Several methods exist to process the information provided in a decision environment. These methods can be classified according to the decision maker's preference information. Hwang and Yoon make this classification in three stages, provide a taxonomy to aid in method selection, and present an extensive overview of several methods. Chankong and Haimes present methods and an excellent introduction to the theory. Keeney, alone, and in collaboration with Raiffa, has published extensively in the field. Appendix C contains a review of Multiple Attribute model theory, methods, and a suggested bibliography.

III. INFORMATION REQUIREMENTS

The intelligence phase of the decision making process consists of the following:

1. identifying organizational goals;
2. defining tasks required to meet the goals;
3. gathering the data necessary to accomplish the task;
4. classifying the task according to structure. [Ref. 2]

For the purposes of this paper, it is assumed that organizational goals have been previously defined by an authority higher than the decision maker, and the task of weapons systems acquisition has been assigned in support of those goals. Following standard economic thought, managerial decisions are evaluated on the basis of costs versus benefits. Acquisition costs could include concept exploration, demonstration and validation, full scale development, production, and/or deployment. These costs could include research and development, procurement, O&M and/or life cycle costs. Benefits in this case, are the increased capability or survivability of the weapons systems or personnel given the particular development option or level of investment is available. The specific task assigned to the decision maker is to decide which development programs will be funded to provide the most effective weapons systems within given funding constraints. With these considerations in mind, the data necessary to complete the task is sought.

In order to make responsible, informed acquisition decisions, senior officials must have a variety of data and information available. The most basic elements of this information are:

1. the major characteristics of the weapons systems which comprise the particular warfare area;
2. the development options (investment levels) available for each major system;
3. the cost of each development option;
4. the expected increase in capability derived from each option;
5. how each system ranks in contribution to attaining the desired goals of the specific warfare area, in relation to the other systems;
6. prevailing budget constraints, both for the warfare area, and for the component weapons systems, if applicable. [Ref. 15]

The existing infrastructure of the U.S. Navy already provides this information. To be truly effective, the senior decision maker must have all this information at his fingertips. Armed with all the necessary tools, he is able to chose which development projects will be funded. The justifications and rationale for the assessments which produced the aforementioned information can be judged on their own merits. The decision maker must be given the opportunity to assess the impact of varying these basic data elements.

Given the inputs, a system to support the required decision must provide the decision maker with enough information to complete a rational, economically sound decision. The decision maker needs total costs and overall benefits realized from each possible mix of the available

11

weapons systems options. Each possible mix, created by choosing one option (investment level) from each of the weapons systems in the warfare area, is called an architecture. Total costs are simply the sum of the costs of each option in the architecture under consideration. Overall benefit must be derived using some form of multi-criteria decision making strategy. Since the recommended architecture is merely the result of numerical calculations, the decision maker must have available some sort of "what if" capability to be able to seek out the best mix. The decision maker reviews the utility assessments and may make justifiable modifications. The architecture with the highest overall utility, which falls within the required budget constraints, is the optimal candidate system for additional consideration and/or adoption.

The decision environment can only be characterized as semi-structured at best. Therefore, a decision support system becomes invaluable to the decision maker. For example, the decision maker, sensitive to the political realities of his decision, may be forced to consider other architectures in order to comply with those realities. The optimal solution may not always seem the best in terms of political acceptability. However, a properly supported DSS can be the basis of reevaluating the political aspects of the task. The

method employed by this DSS could be used to prove the infeasibility of certain architectures favored for political reasons [Ref. 16].

Politics aside, the sheer complexity of the task lends itself to machine support. The number of possible warfare architectures is the product of all options in every weapons system category. As the options increase, the total number of architectures increases quite rapidly. For example, with six weapons systems categories, and five options in each, there are 15,625 architectures possible! No human decision maker can possibly consider all those architectures without machine support.

IV. DECISION SUPPORT SYSTEM DEVELOPMENT AND DESIGN

A. DEVELOPMENT

The design of a decision support system should closely mirror the decision environment and the decision style of the decision maker. Relevant questions prior to developing a DSS inquire into the following general areas:

1. Application Theory -- Why is this system required? Who going to use it? How is it going to be used? What solutions will the system provide?

2. Concept -- How will the system work? What is the system's approach for solving the problem?

3. Representations -- What information will the system need to represent to provide the solutions and to support the solution concept? What are useful internal and external representations for this information?

4. Operations -- What commands and operations will the user need to execute in order to obtain the solutions? [Ref. 6]

1. Application Theory

Chapter I introduced the background and requirements for this DSS. Weapons systems development projects involve several hundred millions, often billions of dollars. Any tool to aid decisions of which projects are worthy of investment can improve effective use of limited funds with significant savings. Different development projects offer several avenues. Options range from entirely new weapons systems to routine maintenance of existing systems.

14

In light of increasing pressure from the Congress, the Department of Defense is continuing to develop its joint operating capabilities. Future weapons systems will not have the luxury of operating in the relative isolation of one warfare area or even one service. Since all government funding comes from the Congress, acquisition and development project decisions cannot be separated from political considerations. These facets must be considered in the increasingly complex and dynamic decision environment in which weapon systems development exists.

In the U.S. Navy, weapons systems acquisition and development decisions are made by senior flag officers supported by numerous military and civilian experts. These experts provide input to the decision maker on future warfare needs, viable options, costs, and the degree of increased capability which various options offer. The decision maker's staff collects this input and combines it with service-wide tasking and strategic plans from the Department of Defense. The decision maker then provides his input with supporting documentation to the Office of the Secretary of Defense as a budget submission.

The process described above lacks a method for the decision maker to pursue different options efficiently. His decision rests heavily on the recommendations of staff and experts. Little capability, other than his own expertise, is provided for the decision maker to test the sensitivity of

the information. An automated decision support system would provide this capability. The decision maker could explore results of varying budget levels, relative weightings of warfare categories, or certain fixed combinations of development options. The resulting budget submissions should include more defensible positions, reached by considering all options available in whatever combinations the decision maker deems relevant.

2. Concept

The DSS should provide a method of presenting all the options available. To make the number of options manageable, only one warfare area should be considered at a time. The system should be capable of evaluating all the option mixes available and displaying the "best" mix according to criteria provided by the decision maker.

Since the ultimate purpose is to provide justifications for budget submissions, the system should use some form of cost versus benefit analysis to identify the best mix of options. This solution will be the mix which provides the greatest benefits, while remaining within established budget limits. Within a limited budget, each option competes with the others for funding. As discussed in Chapter II, decisions in this type of environment lend themselves to Multiple Attribute Decision Making (MADM) methods. (Refer to Appendix C for a more in-depth treatment of MADM methods.) The simple additive weighting method is

16

appropriate in this situation given Hwang and Yoon's taxonomy, illustrated in Figure C-1. In addition, this technique has been successfully used on previous development project decisions and gained acceptance by senior decision makers.

3. Representations

A wealth of information (provided by the staff and experts) exists for each option. This information should supply the decision maker with the justifications and reasoning which assigned the cost, utility, and weights for each option or category of options. The decision maker is then free to consider the validity and relevance of the quantifications.

As discussed in Chapter II, hypertext provides an excellent vehicle to present large amounts of data in manageable quantities. As the decision maker considers the information provided, he may create new links or change existing ones to document his decision process. By paralleling his thought processes, hypertext can provide invaluable support during the decision and aid in decision reconstruction if required.

Externally, the information should be presented in a consistent format which is easy to understand and remember. Data entry and update should be simple and intuitive. Relevant costs, utilities, and weights should be presented for the selected architecture and the individual component

options. Breslawski [Ref. 17] has shown that decision makers exhibit greater satisfaction with the chosen alternative and the decision support system when both types of information are available.

Internally, the data and the solution method should be represented by an appropriate model. In order to describe the model chosen, some definitions and notation explanation are in order. Restricting the discussion to the Anti-Air Warfare area, I will follow the convention of Franck and the U.S. Space and Naval Warfare Systems Command (SPAWAR) by dividing this warfare area into five categories: Surveillance and Warning; Force Coordination; Air Superiority; Battleforce Area Defense; Ship Self Defense. [Ref. 18] Using the simple additive weighting method (SAW), each category is assigned a weight relative to the other categories. These weights may be normalized, however, previous experience with the method at SPAWAR and the Naval Air Systems Command (NAVAIR) has shown greater acceptance without normalized weights.

Within each category, several development options may be presented. For example, the Air Superiority category has six option components: F-14A; F-14A+; F-14D; F/A-18C/D; F/A-18E/F; Next Generation Fighter (NGF). Each option will have a cost associated and a utility or benefit relative to other options within the category. By convention, a utility

score of zero is assigned to the current capability or status quo, and a score of 100 is assigned to the best feasible option.

In a manner similar to the standard "knapsack" problem of Operations Research, one option is chosen from each category to create an architecture. The architectures so created are evaluated against one another in terms of total cost and overall utility. Total cost is the sum of the individual costs for each option, selected from each category, that comprise the architecture. Overall utility for an architecture is defined by the SAW method as the sum of the products of each option utility and weights divided by the sum of the weights, Equation 1. The "best" architecture is defined as the greatest overall utility whose total cost remains within a predetermined budget constraint.

Mathematically, this model can be represented as a linear programming problem. Each category represents a row of a matrix. The columns of the matrix represent the options available in a category, i.e. option(i,j) would be the jth option of category i. Given n categories, a vector of weights $(w_1, w_2, w_3, \ldots w_n)$ must be created to represent category weighting factors. Allowing m options, two n x m matrices represent all available option costs and utilities.

$$
\begin{array}{ccccccc}
u_{11} & u_{12} & u_{13} & \cdot & \cdot & \cdot & \cdot & u_{1m} \\
u_{21} & & & & & & & \cdot \\
\cdot & & & & & & & \cdot \\
\cdot & & & & & & & \cdot \\
\cdot & & & & & & & \cdot \\
u_{n1} & & & & & & & u_{nm}
\end{array}
\quad = \text{utility matrix}
$$

19

$$\begin{matrix} c_{11} & c_{12} & c_{13} & \cdot & \cdot & \cdot & \cdot & c_{1m} \\ c_{21} & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ c_{n1} & & & & & & & c_{nm} \end{matrix} = \text{cost matrix}$$

The linear programming problem becomes integer programming in which options are represented by a doubly subscripted, binary variable, x_{ij}, where $x_{ij} = 1$ indicates the option is selected and $x_{ij} = 0$ means it is not. The evaluation function represents a maximization of overall utility:

$$\text{maximize} \quad \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} w_i u_{ij} x_{ij}}{\sum_{i=1}^{n} w_i} \qquad \text{Equation 1}$$

This function is subject to the following constraints:
for a given category i, i ϵ (1,2,....,n),

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \text{Equation 2}$$

i.e., no more than one option may be chosen
from any category;

$$\sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \leq \text{MAXBUDGET} \qquad \text{Equation 3}$$

i.e., the total cost of an architecture
must be no more than the predetermined
maximum budget constraint, MAXBUDGET.

Non-negativity of x_{ij} is assumed by its definition as a binary variable.

Using this model, an appropriate solver can be chosen. The system will then present the solution to the decision maker for evaluation.

4. Operations

The DSS should provide for consistent and intuitive data entry. Justification and general option information should be accessible to the decision maker. Basic word processing and text features should be available with the justification data to enhance its utility.

"What if" capabilities should be provided for the decision maker to test data sensitivity as discussed in Chapters II and III. The system should allow the decision maker to vary any data element. Automatic recalculation of the solution should occur once an update is entered. Transitions from data to justification information should be immediate and simple to accomplish [Ref. 17]. Update of existing data should be intuitive and consistent with initial data entry.

B. DESIGN

Sprague and Carlson developed a design framework for DSS which they called ROMC. This framework divides design into four categories: Representations; Operations; Memory Aids; Control Mechanisms (ROMC). [Ref. 7]

Hypercard is an excellent generic hypertext engine and encompasses strongly supported graphics capabilities. The designer is limited only by the boundaries of imagination.

Customized, dynamic graphics can be created by the designer or the decision maker to make the system more closely mirror the decision process. Akscyn, et al, develop an extensive set of design issues for Hypermedia systems. [Ref. 19] These issues, in conjunction with Sprague and Carlson's ROMC framework guided the design decisions made for this DSS.

1. Representations

Nodes in the DSS are represented by frames or cards. The primary card introduces the system and establishes a hierarchy among the remaining cards. Refer to Appendix D, Figure D-1. Command options are presented as button links to the separate components of the system: decision matrix; data entry; data information and justification. Unique backgrounds are provided for each of these subdivisions as a visual cue for user navigation.

The decision matrix presents all available options. Corresponding to the model design, rows in the matrix are named for categories. Columns are marked as Option 1, Option 2, etc. indicating increasing benefits, costs, and consequently, increasing risk. Each solution architecture is presented by highlighting the options which comprise it in reverse video, and displaying its total cost and overall utility. (See Figure 4-1.) Figure D-2, Appendix D is a representative decision matrix.

INCREASING BENEFITS, COSTS, AND RISK LEVELS

CATEGORY	option 1-1		option1-2		option 1-3		option 1-4		option 1-5		option 1-6	
	cost		cost		cost		cost		cost		cost	
	wt	utility	wt	utility	wt	utility	wt	utility	wt	utility	wt	utility
weapons system category1												
weapons system category2												
weapons system category3												
weapons system category4												
weapons system category5												
weapons system category6												

Figure 4-1

23

In the data entry component, each option within a category has its own card, all presenting the same background. The cards are, in turn, enlargements of their corresponding entries within the decision matrix. Each card displays the weighting factor assigned to the option's category, plus the option's utility, cost, and name. All cards are full-screen images. Figure D-4 illustrates a typical option card.

The justification and background component consists of cards with scrolling fields which contain the supporting text. Cards are grouped by category, with each card representing one option. Navigation links are provided for access to all options within a category. A find facility permits word or phrase search within the text field. Full word processing capabilities exist to edit the text.

Button links are provided to allow navigation to all system components. The components were purposely limited in number. This strict limitation reduces cognitive overhead and allows more intuitive navigation. Links are provided in two types, hierarchical, and annotation or referential links. Refer to Appendix B for discussion on link types. Hierarchical links exist on the introduction card and between components. These links include icons and the destination name or description. All hierarchical links are components of the cards in which they appear. Their sources are the icon graphics. All link destinations are cards. Most hierarchical

links include internal structures containing HyperTalk scripts. These scripts provide navigation, card structure, and execution of external code resources.

Referential links exist between cards of text within the justification component and between a few button links to create the desired visual effects. All sources and destinations for these links are cards.

All links are executed by the mouse point-and-click action. Icons and descriptive names were chosen to represent links because the vast majority are not simply connections between pieces of text.

The decision matrix presents all available options. Each option in the matrix contains its name, cost, utility, and weighting factor. Corresponding to the model design, rows in the matrix are named for categories. Columns are marked as Option 1, Option 2, etc. A solution architecture is presented by highlighting the options which comprise it in reverse video, and displaying its total cost and overall utility.

2. Operations

Data input is initialized by clicking on the corresponding button link. The user is presented with a series of dialog boxes and the option card representation for data entry. As each card is completed, its corresponding entry in the decision matrix is filled. This action enhances the user's orientation, especially when several options must be entered.

"What if" analysis is performed by clicking a similar button link. The decision maker is asked if a data element (cost, name, utility, or weight) or the budget figure is to be changed. Once selected, the decision maker is prompted for the specific option involved. The corresponding card appears, the matrix is changed accordingly, and a new solution is generated, if required. Corrections to data elements are accomplished via the same mechanism, although the process is begun with a different button link.

Data manipulation to generate solutions is performed by a separate program written in C which HyperCard calls as an external resource (XCMD). The XCMD is executed through scripts written in HyperCard's internal language, HyperTalk. It creates all possible architectures, sorts for the greatest utility within the budget constraint, and returns the solution architecture to HyperCard. The solution's utility, cost, and components are displayed on the decision matrix.

3. Memory Aids

Once data entry, correction, or "what if" analysis is initiated, the solution procedure is called automatically and the new solution is displayed. This design feature relieves the decision maker from extra manual effort and requires no memorization of the solution procedure. The design assumes a solution is the ultimate goal.

To assist in navigation and user orientation, the three major components of the system (decision matrix, option

data, and option justifications) all exist on different card backgrounds. Access to the introduction card is provided on all component cards.

The majority of cards exist in the data component. These cards have the associated option and Category name to enhance orientation. Button links to adjacent option cards are provided for navigation.

4. Control Mechanisms

All internal structures of button links consist of HyperTalk scripts. The hierarchical links create a form of menu by limiting navigation to different components of the system.

All data entry is provided through standard dialog boxes. This feature ensures proper entry and makes the data available for use by various scripts. Execution of the solution XCMD is only available through scripts.

HyperCard provides several user levels which can be set by the designer. The designer is given the option to allow a user to change his access level. User access levels range from merely browsing to full command of the system. Full command entails creating or altering any object or its properties. This power includes access to object scripts and their alteration. Intermediate levels provide varying access to system objects.

Standard Apple menubars may be shown or hidden by the author or designer. These menubars allow increased access to

individual parts of the system, file operations, etc. A message box is provided by HyperCard (normally not visible) which can be used as a form of command line program control. Password provisions can be included as well.

As a prototype system, user access is presently unlimited. Final implementation should restrict the decision maker to the authoring level. This level allows the creation of new objects (cards, button links, or text fields) but denies access to designed scripts and code.

V. CONCLUSIONS AND RECOMMENDATIONS

The research questions addressed in this study were presented in Chapter I. This paper has presented a method to answer those questions in the form of a decision support system.

A. CONCLUSIONS

What information is required for senior warfare decision makers to reach a best fleet mix? Originally addressed in Chapter III, this question could also be stated:

"What information is required to support a budget submission to Congress to fund a particular fleet mix?"

The Department of Defense's budget justifications continue to be cost versus benefit analyses. Cost is always the primary driving force in development projects. Cost must be balanced by expert military and civilian determinations of the expected benefits. Individual projects must be weighed against competing projects. During downsizing eras and increasing budget deficits, competition for funds becomes increasingly fierce. Warfare areas and their components also compete for a shrinking budget. This DSS provides one method to compare relative benefits against costs in an attempt to identify a "best" mix. The information requirements discussed in Chapter III, political realities, and technological capabilities must all be considered as decision variables.

How should this information be presented to the decision maker? Given the complex decision environment described in Chapter III, an interactive decision support system built on a hypertext vehicle can become an invaluable tool. Chapter II delineates the advantages of hypertext compared to standard linear presentation methods. Hypertext's capability of presenting large amounts of data in manageable chunks is its outstanding feature for this decision environment. Rapid support of inter-document linking and the excellent graphics capability of HyperCard establish it as a premier development platform. Both individual option and architecture attributes are readily available to the decision maker. Sensitivity and "what if" analysis is easily performed. Access to expert opinion is instantly available to aid in forming a rational and effective decision.

What method should be used to synthesize the raw data to produce the required information? The decision environment lends itself to Multiple Attribute Decision Making methods. Chapter II presents the characteristics of decisions in which these methods are appropriate. Multiple goals, conflicting options, and a finite set of alternatives are the primary considerations in choosing Multiple Attribute Decision Making methods.

The simple additive weighting method (described in Appendix C) was chosen both for its applicability and simplicity. Most option attributes are readily quantifiable.

Keeping the set of attributes to a small number reduces the decision maker's cognitive overhead. In addition, this method has enjoyed previous use and acceptance by high level decision makers in both the U.S. Navy and the Department of Defense.

B. RECOMMENDATIONS FOR FURTHER RESEARCH

Develop an interface with existing mainframe applications. The Operations Research Department at the Naval Postgraduate School has developed a mainframe application written in GAMS which evaluates several budget constraints simultaneously. An interactive, graphical interface to this application would be a valuable decision aid.

Develop a similar decision support system for an IBM-compatible microcomputer. At present, suitable IBM compatible hypertext development tools are nonexistent. When a such a tool is available, a decision support system similar to the one developed in this study would be very valuable considering the heavy investment in IBM-compatible microcomputers.

Develop the external resource in ADA. This action would conform to Congress' mandate that all Department of Defense software development be coded in ADA.

Expand the existing DSS to incorporate a larger data set and improved user interface. The present prototype accommodates six categories with six options each. Expansion of this capability would create a more general tool. Present

31

HyperCard limitations can be overcome through customized menus, color graphics, and customized dialog boxes.

C. SUMMARY

Weapons systems development and procurement exist in a complex and dynamic environment. The decision support system described in this thesis can provide invaluable assistance to a decision maker in that environment. The DSS couples an easily understood interface with a proven and accepted solution method. More than ever, the Department of Defense and the U.S. Navy must present a coherent, defendable weapons systems acquisition strategy. This DSS can form the basis for that strategy.

APPENDIX A

I. DECISION SUPPORT SYSTEMS

A. DECISION THEORY

According to Simon, decision making involves three phases:

1. intelligence--recognizing a decision is required and gathering the necessary data;
2. design--deciding on a course of action and synthesizing the data into information;
3. choice--choosing a result based on the information presented. [Ref. 2]

Some common decision making strategies are: optimizing; satisficing; sole decision rules, selection by elimination; incrementalism. Optimizing involves selecting the alternative with the highest payoff. This strategy requires detailed cost and benefit data and often applies to structured decision problems rather than unstructured problems. Decision makers rarely use optimizing unaided because of the large volume of data required and the time spent in calculation. The optimization strategy is an ideal candidate for automation. In the absence of automation, decision makers often disregard some alternatives or place too much emphasis on intangible, non-quantifiable aspects in order to reduce the volume of data.

Satisficing involves setting minimum standards and choosing alternatives that meet them, i.e. a "good enough"

33

solution. Multi-criteria decision making techniques are often used, but the overriding consideration is the "satisfactoriness" of the solution. [Ref. 2]

The basis for satisficing is the limited human capacity for processing data. A satisficing strategy may also be chosen to limit the cost of decision making or to meet strict time constraints.

Decisions can be made on the knowledge of experts, often referred to as sole decision rules. A variation of this strategy is relying on a single method or data set to formulate a decision. Implusive decisions and those decisions made under extreme time constraints often fit in this category.

Selection by elimination involves ranking decision criteria and establishing minimum standards or ranges for each. Alternative which fail to meet the most important criterion are eliminated until every alternative has been considered. The elimination process is continued with the remaining alternatives considering the next highest criterion, etc., until all criteria have been satisfied or only one alternative remains.

This strategy has certain pitfalls. The decision maker may run out of alternatives rather early in the process, or end with too many. The elimination process is entirely dependent in the ranking of criteria and the thresholds considered as satisfying them. Some alternatives which are

34

actually "better" may be eliminated early on. The ranking of criteria and threshold setting can quite often be fraught with politics and personal agendas.

Incrementalism is useful in situations where the desired result is very difficult or cannot be quantified. This strategy entails a recursive type of satisficing which progressively approaches a "goal". As the process continues, goals, or criteria may change. The strategy is useful in highly unstructured decision problems.

Selection of a decision strategy is driven by the basic characteristics of the decision environment:

1. scope of the decision--individual vs organizational focused;
2. nature of the decision maker--individual vs group;
3. impact of the decision--inexpensive-to-change vs expensive-to change;
4. time available to make the decision;
5. degree of structure the decision problem presents. [Ref. 6]

B. DSS MODELS

The Sprague and Carlson design for generic decision support systems consists of three management components: data; models; dialogues, or user interfaces. [Ref. 7] The data management component houses all the facilities necessary to edit, retrieve, store, and delete the data required by the decision support system. It contains all the basic subsystems considered essential in database management system: a data

dictionary for meta-data and data; a query system; data security facilities; usage audit facilities, in addition to data manipulation.

The model management component provides similar functions to manipulate and manage models. This component provides facilities to integrate, solve, and validate models. The management system allows update and retrieval of all models stored in the model base. Security and audit features are also available for each model.

The dialog component controls all of the interaction between the user and the other components. It consists of menus, languages, and control mechanisms which allow user access to the data and models in the decision support system. The dialogue component may have a natural language processor as an interface between internal languages. Interfaces with peripheral devices are included to provide a means to display data and solutions. The dialogue component incorporates help facilities and error messages as a part of the interface with the user.

Bonczek, Holsapple, and Whinston described a similar design for decision support systems. Their description also consists of three components: a language system; a knowledge system; a problem-processing system. [Ref. 8]

The language system component (LS) consists of all the linguistic facilities that exist between the DSS and the decision maker. The LS is the user interface in a

computerized DSS. Just as humans are limited by language barriers, the ability of the decision maker and the DSS are likewise limited. Both participants must express themselves in a common language. Thus, the LS sets limits on the interactions of the decision maker and the DSS.

The knowledge system (KS) consists of the facts specific to the problem domain. These facts may be data or models or both. The majority of the power and utility of DSS resides in the KS. The facts stored in the KS not only provide the basis for a solution, but alternative solutions, justifications, and metrics to evaluate the effectiveness of each solution.

The problem-processing system (PPS) serves as the interface between the LS and KS. The PPS is the heart of the DSS. Within the PPS resides the solver for the model in the KS. The PPS also contains one or more of the seven abilities required by a decision maker.

Decision makers possess seven general abilities. These abilities were proposed by Bonczek, et al, in two postulates. The first postulate states that there exist three aspects of decision makers: power; perception; design. None of these aspects can be expressed in terms of the others.

> Power refers to directive force, the ability to govern
> and govern and to eliminate that which is unresponsive.
> Perception includes vision and insight: it is the ability
> to observe, to gather information. Design refers to the
> ability to formulate (e.g. to formulate models). [Ref. 8]

The second postulate states that the existence of the three basic aspects implies four additional aspects which are

37

unique mixtures of the original three. These are: analysis; idealism; implementation; adaptation.

Analysis is the combination of perception and design. It is the continuing meditation between perceptions and formulations, between gathered information and models for processing information. Analysis results in beliefs, knowledge, or expectations.

Idealism is the continued application of power toward a perceived goal. Thus, it is the combination of power and perception and its result is the promotion of values or ideals.

Implementation is the execution of a plan or coordination of an activity according to some plan. As such, implementation is the coordination of power and design [Ref. 8].

Adaptation is the interaction and adjustments made among all three basic aspects and the corresponding secondary facets proposed by the second postulate.

> Given conflicts in or alterations in the available powers, perceptions, and designs, adaptation refers to the struggle within the decision maker to create an equilibrium. Since this fact involves mediation of the three basic facets, it also involves adjustments among the three facets that are pairwise derivatives of the three basic facets; in other words, the adaptation facet is the adjustment process among the other six facets. [Ref. 8]

Adaptation is the heart of the effective decision maker. It provides him the ability to recognize the requirements for decisions and make them to resolve problems. Figure 2-1 illustrates the relationship between all seven facets.

These seven abilities provide a method of designing and evaluating a DSS. The number of abilities which are automated in the DSS and the degree in which they support the decision maker provide a measure of the DSS "intelligence."

All of the abilities cannot reside within the DSS. This fact forms the basis of the system being designed as a support tool. The DSS cannot replace the decision maker because it cannot simultaneously embody all seven facets necessary for the decision to be made [Ref. 8].

APPENDIX B

I. HYPERTEXT

A. HISTORY

The first description of hypertext is credited to Vannevar Bush, President Roosevelt's Science Advisor, from his article "As We May Think", written in 1945. Bush's article described a machine he called the memex which would be used to organize and mechanize scientific literature. Bush's primary vision for the memex was for it to become a mechanical memory to support the researcher's thought processes. [Ref. 12]

> The human mind ... operates by association.... One cannot hope to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage. [Ref. 20]

Bush's concepts drove early research in hypertext which developed literary systems such as Englebart's NLS/Augment and Nelson's Xanadu. Other application areas of hypertext research have produced systems in three other general categories: problem exploration tools that support early unstructured thinking; browsing systems, smaller literary systems designed specifically for ease of use; general hypertext designed primarily for development. [Ref. 12]

B. MODELS

Database objects are often associated with windows on the screen in a one-to-one correspondence. Standard window operations such as opening, closing, resizing, and repositioning are supported. The windows may contain any number of links representing pointers to other windows. The user has the ability to create new links to new or existing nodes. The network can be browsed using three common methods: following each link successively; searching for keywords or phrases, much like any database search; using the browser, a tool which represents the database in a graphical form that allows structural navigation. [Ref. 12]

General hypertext systems display many similarities. Research models have been proposed to describe hypertext architectures, based mainly on the concept of successively deeper levels. The Dexter model, proposed by the Dexter Group, consists of three levels with two interfaces between them. The first level is the runtime layer. This level is what the user sees, and defines what interactions are provided by the system. The next level is the storage layer, which contains the database particulars. Between them lies the presentation specifications. The deepest level of the Dexter model is the within-component layer. The basic components, nodes and links, of the hypertext system reside in this layer. Between the storage and within-component layers is the anchoring interface.

41

Campbell and Goodman proposed a similar model of three layers without distinguishing interfaces. Their model is remarkably analogous to the Bonczek-Holsapple-Whinston model of decision support systems.

At the base of this model is the database level. This level maintains the facilities for storage, retrieval, and update of the component objects. Its operation is similar to those of any other general database. This layer contains the information necessary for efficient operation on the objects. Any facilities for multi-user access, and data security will reside in the database level. [Ref. 13]

The next level in the Campbell-Goodman model is the hypertext abstract machine (HAM) level. The HAM contains the information and structure of each object and how they relate to each other, much like the meta-data of a data base management system.

The highest level is the presentation layer. This layer acts as the user interface for the hypertext system. In this layer, the designer decides how each component will be presented to the user. Limits on the user's interactions with the system are defined. The system could also be capable of dynamic interaction limits, · selected by the user, or programmed by the designer. [Ref. 13]

According to Nielsen, the HAM is probably the best level to connect different hypertext systems for data exchange. The database level is generally strongly tied to the machine in

an effort to make it more efficient. Thus the corresponding database levels of two hypertext systems would contain far too many incompatibilities. The presentation level is usually much too varied between hypertext systems to allow interchange. The HAM, being the interface between the database and the user interface becomes the default interchange level. Research in data interchange, conducted mostly in workshops run by the National Institute of Standards and Technology, have produced more detailed architecture models of hypertext systems.

C. COMPONENTS

1. Nodes

The two fundamental components of hypertext systems are nodes and links. Nodes are where the information in a hyperdocument is stored. Nodes tend to be text, although there are no requirements that they be. They may be graphics, sound, or video. If such is the case, the hyperdocument involved is more properly termed hypermedia. Regardless of the form a node takes, it usually expresses only one idea. This fact "invites the writer to modularize ideas into units...." [Ref. 12]

The concept creates both advantages and disadvantages. The major advantage is that nodes more closely resemble human thought processes. Humans reason by ideas and naturally separate them in their minds. Hypertext provides a machine-supported vehicle to support the thought process.

The ability to present modular ideas does create some drawbacks. The reader of a hyperdocument is not constrained to the flow of the writer's ideas as in normal linear text. The reader is free to pursue whatever links to other nodes he wishes. As a result, the ultimate purpose of the writer may become lost. This danger becomes especially apparent when the reader can create his own links. [Ref. 12]

Nodes are often typed to differentiate ideas and to establish some form of hierarchy. The node type is generally made apparent by graphic attributes which are common to all nodes of that type. These attributes may be colors, specific icons, backgrounds, or unique presentation shapes.

Many hypertext systems provide the ability to enforce a structure on nodes. These nodes may consist of separate text fields or spaces for data entry. Structured or semi-structured nodes are often used to enforce requirements that certain facts must occur together. [Ref. 12]

Lastly, similar or related nodes can be grouped in a sort of super-node or composite node. This construction, again, enforces a form of hierarchy within the system.

2. Links

Links are the essence of hypertext. Links generally occur in three types: referential; organizational; keyword. [Ref. 12] Referential links support the reader by providing access to text files related to the document being read. As such, referential links are not hierarchical. These links

44

have two ends and are usually directed, but may be bidirectional. The source of a referential link is generally a point or region in the reference document. The link's destination may be either a point or an entire file. The hypertext system must display the existence of a link to the reader. This may be accomplished by descriptive icons or special fonts within the text. The reader then causes some action, e.g. clicking the mouse, to execute the link.

Organizational links exhibit the same characteristics as referential links, only their purpose is to implement a hierarchy within the document. Typical examples include tables of contents, page turning buttons, or any designs supporting traditional linear text structure. Many hypertext systems provide special internal commands to implement organizational links. These commands exploit established tree hierarchies to make processing more efficient. [Ref. 12]

Keyword links are a form of search which doesn't require explicit action by the designer. These links normally have points for sources (the keyword) and regions for destinations (the found keyword and its surrounding text). The keyword link provides a mechanism to search every node dynamically. These links give the reader much more freedom to customize his research. The hypertext document designer is released from creating a multitude of dedicated links in anticipation of every reader query.

APPENDIX C

I. MULTIPLE CRITERIA DECISION MAKING

A. COMPONENTS

Most multiple attribute decisions consist of five common elements: a decision maker or unit; the decision maker's objectives; certain measurable attributes of those objectives; a decision statement; a decision rule. [Ref. 4] The decision maker need not be a single person, as long as the unit/group can accept a common, unified course of action. The decision maker will receive input data in support of his stated objectives. These data are normally in the form of alternatives, attributes, or both. Using this data, the decision maker manipulates and processes it into a suitable form of information with which he can make a decision, or particular course of action.

Objectives are statements of what the decision maker wants to achieve. These objectives usually exist in some form of a hierarchy. Objectives are classified as operational and non-operational. Operational objectives exist at the lowest levels of the hierarchy. These objectives permit practical methods of measuring levels of achievement.

Attributes are the measurable quantities assigned to each operational objective. These attributes should be

comprehensive, and directly measurable.

The decision situation is a complete description of the problem structure and the decision environment. It will describe the types and number of inputs. The decision situation identifies the decision variables, attributes, and the measurement scales employed. The situation will include any relationships between variables and attributes, and a complete listing of all alternatives.

The decision rule is the yardstick used to measure alternatives. It will provide a ranking of all alternatives in accordance with the defined goal mechanism. The decision rule will normally be a mathematical model which assigns values to each alternative to provide the subsequent ranking. [Ref. 4]

B. MODELS

Noncompensatory models are those which do not permit trade-offs between attributes. Disadvantages within one attribute are not allowed to be offset by greater advantages within another. These models yield fairly simple solvers and are suitable for decisions in which little information about the decision maker's preferences are provided. Representative solvers include minimax, maximin, and lexicographic methods.

Compensatory models do allow attributes to balance each other. Thus, changes in one attribute often can be offset by opposite changes in another. Compensatory models usually employ a single value which the solver uses to rank

47

alternatives. Quite often, this single value will be termed an overall utility. Compensatory models are further subdivided by the method in which the overall utility is assigned. These divisions and their corresponding solving methods are:

1. scoring model -- the alternative with the highest score, or utility is chosen. Representative methods are: hierarchical, simple, or interactive weightings.

2. compromising model -- the alternative which is closest to the ideal solution is chosen. Nonmetric multi-dimensional scaling and the linear programming techniques for multi-dimensional analysis of preference (LINMAP) are methods which belong to this division.

3. concordance model -- the alternative which best satisfies a given concordance measure according to set of preference rankings. Permutation methods and linear assignment are concordance methods. [Ref. 14]

C. SOLVING METHODS

Several methods exist to process the information provided in a decision environment. These methods can be classified according to the decision maker's preference information. Hwang and Yoon [Ref. 14] make this classification in three stages: (1) the type of information required from the decision maker (attribute, alternative, or none); (2) the primary aspect of the information; (3) the major methods which correspond to the elements of stages (1) and (2). Figure C-1 (from Ref. 14) illustrates this classification.

Following the taxonomy provided in figure C-1, the decision maker may have no preference of attributes or alternatives, or may not have enough knowledge to form

48

preferences. The decision method becomes one of selecting the alternative with the highest payoff.

Provided the decision maker has expressed preferences on alternatives or attributes, other methods can be employed to generate solutions. The preferred method is a function of how the attributes or alternatives are ranked and whether trade-offs between them are allowed. Hwang and Yoon [Ref. 14] present an extensive overview of several popular methods.

C. BIBLIOGRAPHY

The following bibliography contains excellent references for further research in this discipline:

Farquhar, P.H., "A Survey of Multiattribute Utility Theory and Applications," M. Starr and M. Zeleny (eds.), Multiple Criteria Decision Making, North Holland, New York, 1977.

Fischer, G.W., "Experimental Application of Multi-Attribute Utility Models." D. Wendl and C. Vlek (eds.), Utility, Probability, and Human Decision Making, D. Reidel Pub. Co., Boston, 1975.

Fishburn, P.C., "A Survey of Multiattribute/Multicriteria Evaluation Theories," S. Zionts (ed.), Multiple Criteria Decision Making: Kyoto 1975, Springer-Verlag, New York, 1976.

Keeney, R.L., "Utility Independence and Preferences for Multiattributed Consequences," Operations Research, Vol. 19, No. 4, pp. 875-893. 1971.

Keeney, R.L., "Utility Functions for Multiattributed Consequences," Management Science, vol. 18, no. 5, part 1, pp. 276-287, 1972.

Keeney, R.L., and Raiffa, H., Decision with Multiple Objectives: Preferences and Value Tradeoffs, Wiley and Sons, New York, 1976.

Type of Information from the Decision Maker	Salient Feature of Information	Major Classes of Methods
No Information		Dominance Maximin Maximax
Information on Attribute	Standard Level	Conjunctive Method Disjunctive Method
	Ordinal	Lexicographic Method Elimination by aspects Permutation Method
	Cardinal	Linear Assignment Method Simple Additive Weighting Method Hierarchical Additive Weighting Method ELECTRE TOPSIS
	Marginal Rate of Substitution	Hierarchical Tradeoffs
Information on Alternative	Pairwise Preference	LINMAP Interactive SAW Method
	Order of Pairwise Proximity	MDS with Ideal Point

Multiple Attribute Decision Making
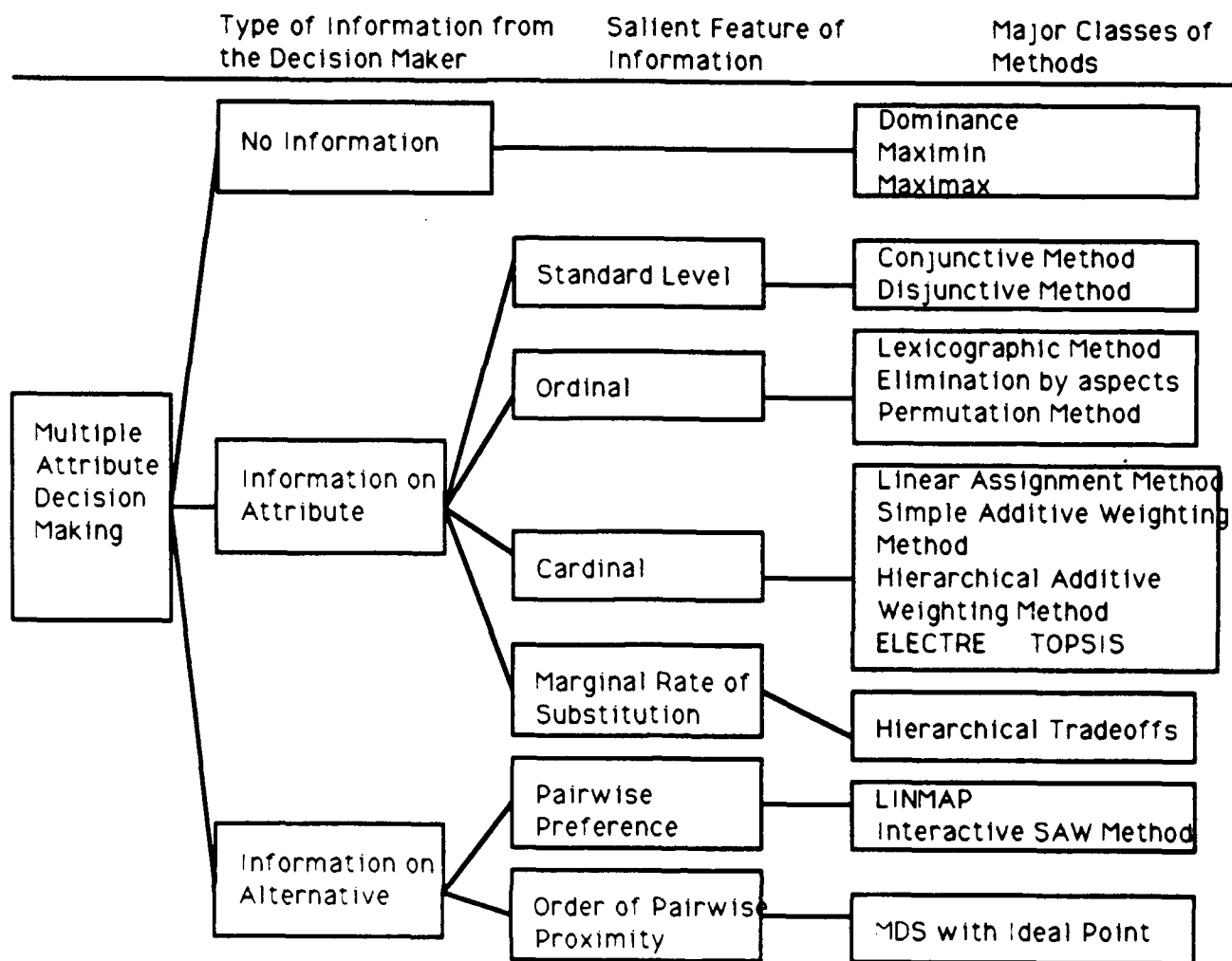
Figure C-1

50

APPENDIX D

I. SELECTED SCREEN DISPLAYS

WEAPONS SYSTEM
DEVELOPMENT
DECISION
SUPPORT
SYSTEM

justifications

data input

matrix

quit

Figure D-1

52
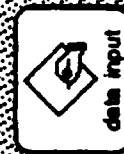
SELECTED ARCHITECTURE		utility		cost		
		33		247		

CATEGORY	OPTION 1	OPTION 2	OPTION 3	OPTION 4	OPTION 5	OPTION 6
F-14 ENGINE	ENGINE1 0 210 0	ENGINE2 2 210 2	ENGINE3 2 210 10	ENGINE4 3 210 12	ENGINE5 5 210 13	ENGINE6 359 210 100
RADAR UPGRADE	RADAR1 0 100 0	RADAR2 0 100 3	RADAR3 0 100 30	RADAR4 162 100 40	RADAR5 364 100 60	RADAR6 497 100 100
ELCTRO-OPTIC AL SENSORS	ELECTRO1 0 50 0	ELECTRO2 5 50 5	ELECTRO3 25 50 45	ELECTRO4 95 50 90	ELECTRO5 160 50 100	
TACTICAL COMMUNICATI ONS	COMM1 0 50 0	COMM2 0 50 0	COMM3 0 50 20	COMM4 0 50 30	COMM5 40 50 65	COMM6 60 50 100
AVIONICS UPGRADE	AV1 0 80 0	AV2 20 80 3	AV3 40 80 9	AV4 60 80 10	AV5 440 80 90	AV6 520 80 100
EW COUNTERMEAS URES	EW1 0 40 0	EW2 10 40 50	EW3 35 40 55	EW4 35 40 60	EW5 50 40 100	

return   data change   what if?

Figure D-2

Enter initial data by clicking
on data input.
View or change data entries
by clicking on options.

data input

options

return

Figure D-3

54

Figure D-4

This card will contain the justifications for the utility, weight, and cost of the first option in category one.

return

next

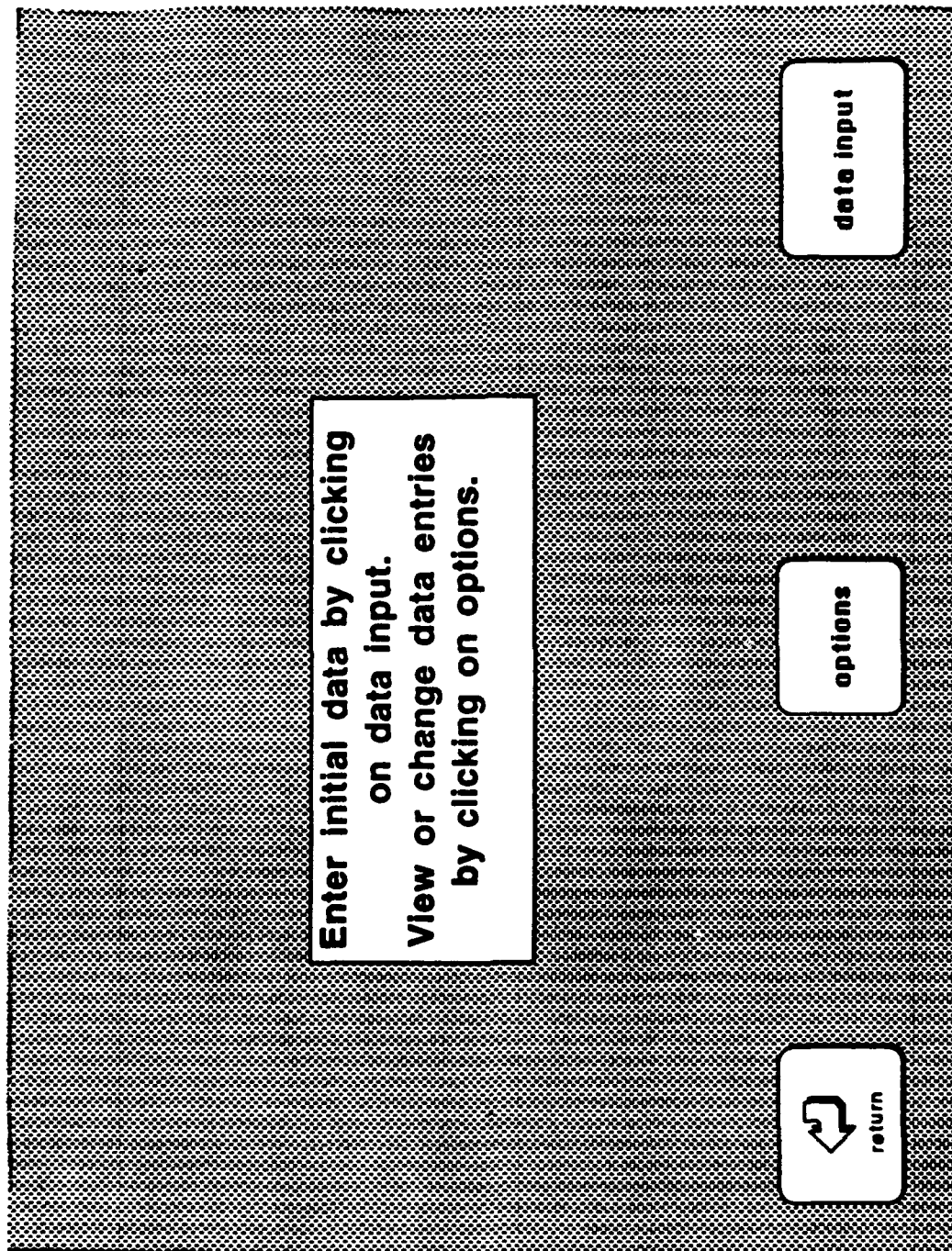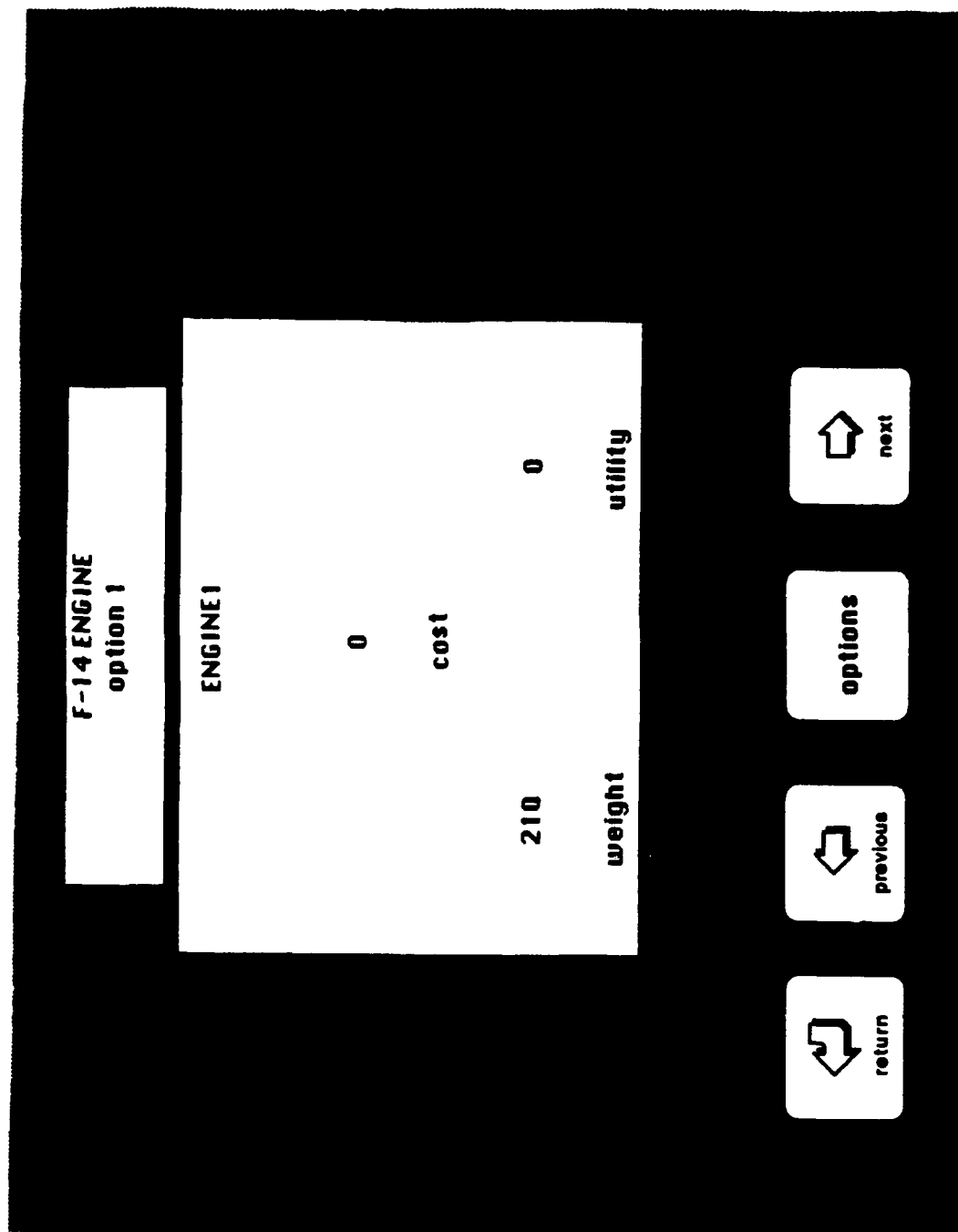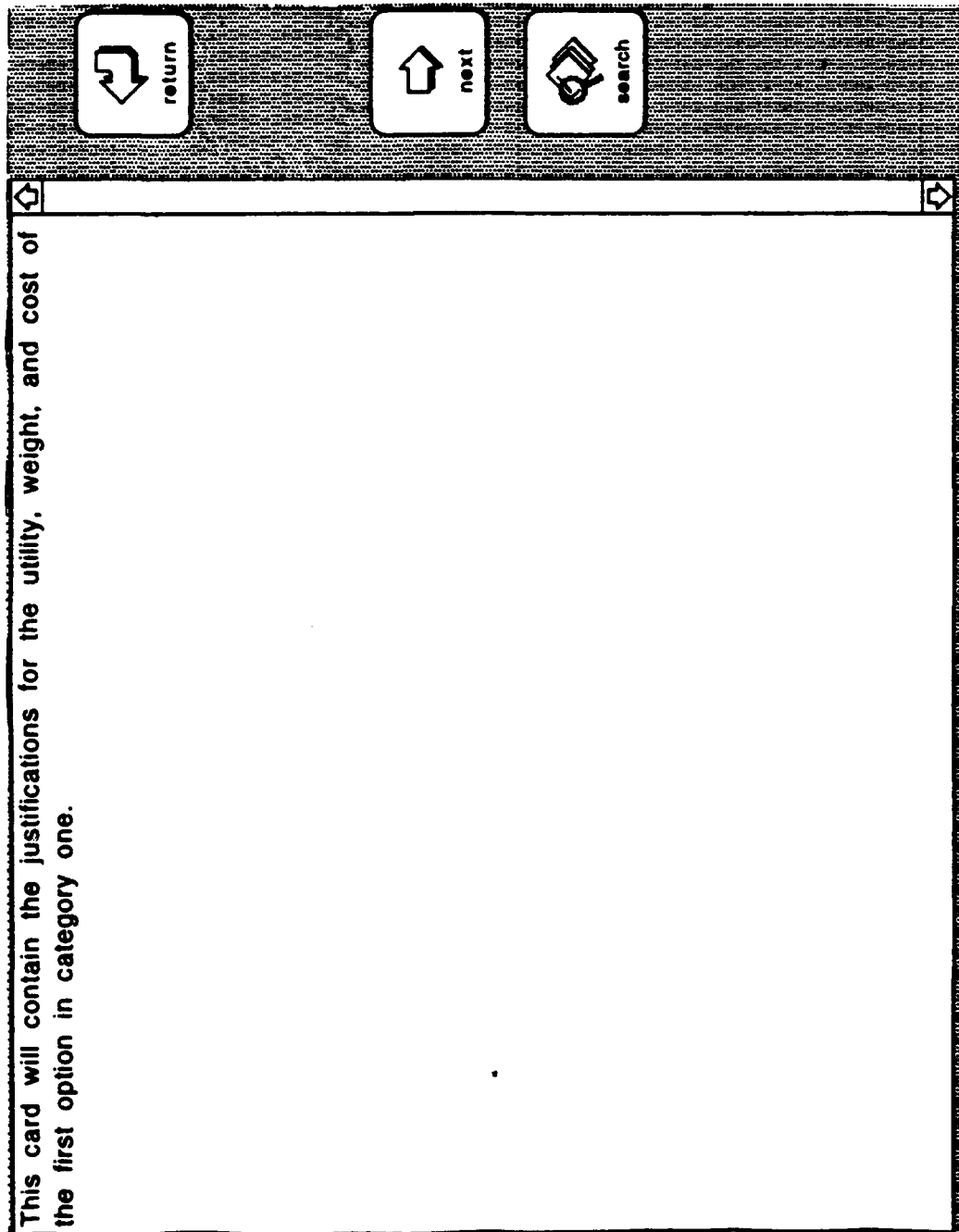search

Figure D-5

APPENDIX E

I. SOURCE CODE FOR NUMERICAL SUBSYSTEM

This source code was written by the author as the numerical subsystem for the decision support system. All of the Hypercard interfaces were taken from suggestions found in , *XCMD's for Hypercard*, [Bond].

/*Richard K. Boyd
1992

makeArch: A HyperCard XCMD written for the stack DSS, as part of a master's thesis from the Naval Postgraduate School. This program creates all possible architectures(combinations) from the data stored in NAMEFILE. NAMEFILE is a text file created by the Handler storeNumbers2 in the DSS stack. The architectures created are written to ARCHFILE. Another handler reads them, selects the requested architecture and displays it on the matrix card of the stack DSS.

Form: makeArch parameter[1] parameter[2] ....parameter[16]

Example: makeArch 2 3 5 3 6 5 maxBudget wtlist costlist
            prodlist row1names row2names row3names
            row4names row5names row6names

Notes: makeArch is called from HyperCard scripts. The parameters are Pascal strings initialized within the script of button "Data Input" in stack DSS. The program converts the Pascal string to a zero-terminated C string. Parameters[1]-[6] are the number of options in each row of the matrix, respectively. Parameter[7] is the maximum budget target. Parameters[8]-[10] are ordered lists of the option data, [8] is the ordered

list of weights, [9], the costs, etc. Parameters[11]-[16] are the lists of option names from each row of the matrix.   */

```c
#include <MacTypes.h>
#include "HyperXCmd.h" /*This file defines the HyperCard
                                interface.*/
#include <stdio.h>
#include<SetUpA4.h>    /*This file sets up jump addresses in the
                                A4 register. */

/*defined constant*/

#define SIZE 6  /*This is the number of rows (categories) */

struct matrix_element {
    char        cost[5];
    char        weight[5];
    char        utility[5];
    char        name[12];

} m[SIZE][SIZE], *ap;

pascal void main(XCmdBlockPtr);

void HandleToCstr(char *, Handle);

struct matrix_element *make_matrix(struct matrix_element
                    n[SIZE][SIZE],
                    int a, int b, int c, int d, int e, int f,
                    char *, char *, char *, char *, char *,
                    char *, char *, char *, char *);

char make_list(struct matrix_element p[SIZE][SIZE], int budget,
            int a, int b, int c, int d, int e, int f);
char *CollectToComma(char *, char *);
```

```
pascal void main(paramPtr)
    XCmdBlockPtr paramPtr;
{

RememberA0();
SetUpA4();

/* define variables */

char * str[SIZE];

char solution;

int x;

/*First convert paramPtr->params[i] to C strings, then to integers
  in the function calls.*/

for (x = 0; x < paramPtr->paramCount; x++)
   HandleToCstr(str[x], paramPtr->params[x]);

/* Make-matrix creates the matrix of option data. */

ap = make_matrix(m, atoi(str[0]), atoi(str[1]), atoi(str[2]),
                    atoi(str[3]), atoi(str[4]), atoi(str[5]), str[7],
                    str[8], str[9], str[10], str[11], str[12], str[13],
                    str[14], str[15]);

/* Make_list creates the architectures from the data, tests against
   the budget target, and returns the architecture with the greatest
   utility, whose cost is less then or equal to the budget target. */

solution = make_list(m, atoi(str[6]), atoi(str[0]), atoi(str[1]),
                        atoi(str[2]), atoi(str[3]), atoi(str[4]),
                        atoi(str[5]));
```

```c
/* These statements put the solution into a form which HyperCard
   can receive. */

paramPtr->returnValue = (Handle) NewHandle((long)strlen(solution))
                                                    + 1);
strcpy((char *) *(paramPtr->returnValue), solution);

RestoreA4();
}

/*This function creates the matrix with the data passed through
   the lists. */

struct matrix_element *make_matrix(struct matrix_element
                    n[SIZE][SIZE],
                    int a, int b, int c, int d, int e, int f,
                    char *wtlist, char *costlist, char *utillist,
                    char *names1, char *names2, char *names3,
                    char *names4, char *names5, char *names6)

{

  char  *util, *cost, *weight, *label, *name[6];

  int i,j, colno, row[SIZE];

  row[0] = a;
  row[1] = b;
  row[2] = c;
  row[3] = d;
  row[4] = e;
  row[5] = f;

  name[0] = names1;
  name[1] = names2;
  name[2] = names3;
  name[3] = names4;
  name[4] = names5;
  name[5] = names6;
```

```c
for (i = 0; i < SIZE; i++) {
        colno = row[i];
        for (j = 0; j < colno; j++) {
            utillist = CollectToComma(utillist, util);
            strcpy(n[i][j].utility, *util);
            wtlist = CollectToComma(wtlist, weight);
            strcpy(n[i][j].weight, *weight);
            costlist = CollectToComma(costlist, cost);
            strcpy(n[i][j].cost, *cost);
            name[i] = CollectToComma(name[i], label);
            strcpy(n[i][j].name, *label);

        utillist++; /* Advancing the pointer jumps over the comma */
        wtlist++;   /* so the next string stripped off doesn't    */
        costlist++; /* include it.                                */
            }
        }
    }


char make_list(struct matrix_element p[SIZE][SIZE], int budget,
               int a, int b, int c, int d, int e, int f)
{

char *tempLine, *lineUtility, *lineCost, *maxLine;

float value;

int i,j,k,l,m,n, colno, row[SIZE], archcost, wt, maxUtility = 0;


row[0] = a;
row[1] = b;
row[2] = c;
row[3] = d;
row[4] = e;
row[5] = f;
```

```c
for (i = 0; i < row[0]; i++) {
    for (j = 0; j < row[1]; j++) {
        for (k = 0; k < row[2]; k++) {
            for (l = 0; l < row[3]; l++) {
                for (m = 0; m < row[4]; m++) {
                    for (n = 0; n < row[5]; n++) {

archcost = atoi(p[0][i].cost) + atoi(p[1][j].cost) + atoi(p[2][k].cost)
        + atoi(p[3][l].cost) + atoi(p[4][m].cost) + atoi(p[5][n].cost);

    wt = atoi(p[0][i].weight) + atoi(p[1][j].weight) +
atoi(p[2][k].weight)
        + atoi(p[3][l].weight) + atoi(p[4][m].weight) +
atoi(p[5][n].weight);

    value = (atoi(p[0][i].utility) + atoi(p[1][j].utility) +
atoi(p[2][k].utility)
        + atoi(p[3][l].utility) + atoi(p[4][m].utility) +
atoi(p[5][n].utility))/wt;

    sprintf(tempLine,"%d,%d,%s,%s,%s,%s,%s,%s", archcost, (int) value,
        p[0][i].name, p[1][j].name, p[2][k].name,
        p[3][l].name, p[4][m].name, p[5][n].name);

    tempLine = CollectToComma(tempLine, lineCost);
    tempLine++;
    tempLine = CollectToComma(tempLine, (char *) lineUtility);

            if (*lineCost <= budget) {
            if (*lineUtility > maxUtility) {
                maxUtility = *lineUtility;
                sprintf(tempLine, "%d,%d%s", lineCost, lineUtility,
tempLine);
                }
            }
```

```
                    } /* n loop */
                  } /* m loop */
                } /* l loop */
              } /* k loop */
            } /* j loop */
          } /* i loop */


return(*maxLine);
}
/* This utility function copies the string pointed to by a handle into
    a C string character array,.*/

void HandleToCstr(str, hndl)
    char * str;
    Handle hndl;
{
  strcpy(str, *hndl);
}


    /* CollectToComma is borrowed from "XCMD's for HyperCard". This
function strips off all characters in the string, targetStr, prior to a
comma, placing them in the string, subStr. */

  char *CollectToComma(targetStr, subStr)
    char *targetStr;
    char *subStr;
  {
    while ((*targetStr != ',') && (*targetStr != 0))
      *subStr++ = *targetStr++;
    return(targetStr);
  }


/*XCmdGluec.c was adapted from "XCMD's for HyperCard". This file
    contains some of the utility routines used in the HyperCard XCMD
    interface.*/

#include "XCmdGluec.c"
```

63

LIST OF REFERENCES

1. *Naval Warfare Publication 1 (Rev A) Strategic Concepts of the U. S. Navy*, Chief of Naval Operations, May 1978.

2. Simon, H., *The New Science of Management Decision*, New York, Harper and Row, 1960.

3. Turban, E., *Decision Support and Expert Systems: Management Systems Support*, New York, Macmillian Publishing Co. 1990.

4. Chankong, V.; Haimes, Y. 1983, *Multiobjective Decision Making: Theory and Methodology*, Elsevier Science Publishing Co., Inc., New York, 1983.

5. Gorry, G., and Scott-morton, Michael S., "A Framework for Management Information Systems," *Sloan Management Review*, Fall, 1971.

6. Bhargava, H.K., *A Logic Model for Model Management: An Embedded Languages Approach*, Ph.d. thesis, University of Pennsylvania, Department of Decision Sciences, Philadelphia, Pa., 1990.

7. Sprague, R.H., and Carlson, E.D., *Building Effective Decision Support Systems*, Prentice-hall, Englewood Cliffs, N.J., 1982.

8. Bonczek, R. , Holsapple, C., and Whinston, A., *Foundations of Decision Support Systems*, Academic Press, New York, 1981.

9. Nelson, T.H., Getting it Out of Our System," in *Information Retrieval: A Critical Review*, G. Schechter, ed., Thompson Books, Washington, D.C., 1967.

10. Kimbrough, S.O., Bhargava, H.K., and Bieber, M., On Knowledge Support and Numeric Programming in a Hypertext Environment," draft for University of Pennsylvania, Department of Decision Sciences, Philadelphia, Pa., August 7, 1987.

11. Marchionini, G., and Schneiderman, B., Finding Facts versus Browsing Knowledge in Hypertext Systems," *Computer*, January 1988.

12. Conklin, J., "Hypertext: An Introduction and Survey," IEEE Computer, Vol. 20, No. 9, (September) 1987.

13. Nielsen, J., Hypertext and Hypermedia, Academic Press, San Diego, Ca., 1990.

14. Hwang, C., and Yoon, K., "Multiple Attribute Decision Making: Methods and Applications," in M. Beckman and H.P. Kunzi (eds.), Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York, 1981.

15. Riddle, C.L., A System Costs Planning Decision Support System, Master's Thesis, Naval Postgraduate School, Monterey, Ca., September, 1988.

16. Interview between Gordon Nakagawa, Adjunct Professor, Operations Research Department, Naval Postgraduate School, and the author, 19 February, 1992.

17. Breslawski, S., "The Presentation of Alternatives in Multiple Criteria Decision Support," Proceedings of the Ninth International Conference on Information Systems, J. Degross, and M. Olsen (eds.), November 30 - December 3, 1988.

18. Franck, S.G., A Methodology for Evaluating Warfare Architectures, Master's Thesis, Naval Postgraduate School, Monterey, Ca., September 1991.

19. Akscyn, R.M., McCracken, D.L., and Yoder, E.A., "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations," Communications of the ACM, Vol. 31, No. 7, July 1988.

20. Bush, V., "As We May Think," Atlantic Monthly, July 1945.

BIBLIOGRAPHY

Arinze, B., "Developing Decision Support Systems from a Model of the DSS/User Interface," Knowledged-based Management Systems, G. Doukidis, and F. Land (eds.), Ellis Howard, Ltd., Chichester, Eng., 1989.

Bhargava, H.; Bieber, M.; Kimbrough, S., "Oona, Max, and the WYWWYWI principle: Generalized hypertext and model management in a symbolic   programming environment," Proceedings of the Ninth International Conference on Information Systems, J. Degross and M. Olsen (eds.), November 30-December 3, 1988.

Bond, G., XCMD's for HyperCard, Management Information Source, Inc., Portland, Or, 1988.

Decision and Designs, Inc., Suite 600, 8400 Westpark Drive, McLean, Va. 22101, A Multi-Attribute Utility Model for Evaluating Alternative Naval Aviation Plans, by Rhees, T.R., O'Connor, M.F., Allen, J.J., Selvidge, J.E., Hoblitzell, C.M., May 1977.

Goodman, D., The Complete Hypercard 2.0 Handbook, Bantam Books, New York, 1990.

Halasz, F.G., "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems," Communications of the ACM, Vol. 31, No. 7, July, 1988.

Stanchev, I., "Decisions and Their Support," First IMACS/IFORS Colloquium on Managerial Decision Support Systems, Manchester, U.K., 3-25 November, 1987, Singh, M.G., Hindi, K.S., and Salassa, D. (eds.), Elsevier Science Publishers B.V., 1988.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center   2
    Cameron Station
    Alexandria, VA.   22304-6145

2. Library, Code 052   2
    Naval Postgraduate School
    Monterey, CA.   93943-5002

3. Commander, Naval Air Development Center   1
    Code 3021
    Warminster, PA.   18974-5000

4. Commander, Naval Air Systems Command   1
    Code 526
    Naval Air Systems Command Headquarters
    Washington, D.C.   20361-0001

5. Commander, Naval Ocean Systems Command   1
    Code 4402
    San Diego, CA   92152-5000

6. Commander, Naval Sea Systems Command   1
    Code 06KR
    Naval Sea Systems Command Headquarters
    Washington, D.C.   20362-5101

7. Commander, Space and Naval Warfare Systems Command   1
    Code 31
    Washington, D.C.   20363-5100

8. Chief of Naval Operations   1
    OPNAV - 81
    Navy Department
    Washington, D.C.   20350-2000

9. Johns Hopkins University   1
    Applied Physics Laboratory
    Naval Warfare Analysis Department
    Johns Hopkins Road
    Laurel, MD   20723-6099

10. Superintendent, Naval Postgraduate School   1
    Operations Research Department
    Code OR/NA
    Monterey, CA   93943

11. Superintendent, Naval Postgraduate School   1
    Operations Research Department
    Code OR/LI
    Monterey, CA   93943

12. Superintendent, Naval Postgraduate School   1
    Operations Research Department
    Code OR/SM
    Monterey, CA   93943

13. LCDR Richard K. Boyd   1
    1111 Jefferson Davis Highway
    Crystal Gateway North
    East Tower, Suite 309
    Alexandria, VA   22202